

What is claimed is:

1. A method for implementing bit-fields for a machine word comprising:
defining a variable corresponding to the machine word;
defining at least one bit-field within the variable, the bit-field having a
first bit and a second bit;
5 for each bit-field, determining a first bit number defining a position of
the first bit of the bit-field within the variable;
for each bit-field, determining a second bit number defining a position
of the second bit of the bit-field within the variable; and
for each bit-field, concatenating the first bit number and second bit
10 number together to form a bit-field definition number defining a location of the
bit-field within the variable.
2. A method as defined in claim 1, wherein the variable includes a
structure, further comprising:
defining the variable in a structure declaration; and
for each bit-field, defining an alias for the bit-field definition number
5 within the structure declaration.
3. A method as defined in claim 1 further comprising:
for each bit-field, determining the first bit number by determining a
left bit of the bit-field; and
for each bit-field, determining the second bit number by determining
5 a right bit of the bit-field.
4. A method as defined in claim 1 further comprising:
selecting one of the at least one bit-field from which data is to be
read;
extracting the first and second bit numbers from the bit-field definition
5 number for the selected bit-field;
generating a mask of the variable from the extracted first and second
bit numbers to isolate the selected bit-field;

using the mask to zero-out all bits of the variable except for within the selected bit-field; and

10 using one of the first and second bit numbers to shift the selected bit-field to form the data to be read.

5. A method as defined in claim 1 further comprising:

selecting one of the at least one bit-field to which data is to be written;

providing a write value to be written into the selected bit-field;

5 extracting the first and second bit numbers from the bit-field definition number for the selected bit-field;

generating a mask of the variable from the extracted first and second bit numbers to isolate the selected bit-field;

10 using the mask to zero-out all bits of the variable only within the selected bit-field;

using one of the first and second bit numbers to shift the write value to the location of the selected bit-field; and

inserting the write value into the selected bit-field.

6. A method as defined in claim 1 further comprising:

selecting one of the at least one bit-field for which a mask is to be formed;

5 extracting the first and second bit numbers from the bit-field definition number for the selected bit-field;

subtracting the second bit number from the first bit number to form a first shift value;

10 providing a mask value having one bit of a first logical value and remaining bits of a second logical value, the first logical value bit being an end bit of the initial value;

shifting the first logical value bit from the end bit of the mask value by the first shift value, all bits between the first logical value bit and the end bit

inclusive being first logical value bits, and the remaining bits having the second logical value;

15 inverting all of the bits of the mask value, the first logical value bits becoming the second logical value bits, and the remaining bits having the first logical value; and

 forming the mask for the selected bit-field by using the second bit number as a second shift value and shifting the second logical value bits from the end bit of the mask value by the second shift value, all bits between the second
20 logical value bits and the end bit having the first logical value.

7. A method as defined in claim 1 further comprising:

 selecting one of the at least one bit-field;

 extracting the first bit number from the bit-field definition number for the selected bit-field by dividing the bit-field definition number by a divisor and
5 discarding any remainder from the division; and

 extracting the second bit number from the bit-field definition number for the selected bit-field by dividing the bit-field definition number by the divisor and discarding any non-remainder from the division.

8. A method as defined in claim 7 wherein the divisor is 100.

9. A method of handling a bit-field for a machine word comprising:

 defining a machine word having a plurality of bits including a bit-field located from a first one of the bits to a second one of the bits in the machine word;

 forming a bit-field definition number having a first portion and a
5 second portion, the first portion formed from a location of the first bit in the machine word, the second portion formed from a location of the second bit in the machine word, the bit-field definition number defining a location and range of the bit-field in the machine word; and

 associating the bit-field definition number with the bit-field in the
10 machine word.

10. A method as defined in claim 9 further comprising:

forming the bit-field definition number by concatenating a number of the first bit and a number of the second bit together, the first portion and the second portion representing the numbers of the first and second bits, respectively;

11. A method as defined in claim 10 further comprising:

retrieving the number of the first bit from the first portion of the bit-field definition number by dividing the bit-field definition number by a divisor and discarding any remainder; and

5 retrieving the number of the second bit from the second portion of the bit-field definition number by dividing the bit-field definition number by the divisor and discarding any non-remainder.

12. A method as defined in claim 9 further comprising:

extracting the first and second portions from the bit-field definition number; and

5 generating a mask of the bit-field in the machine word from the extracted first and second portions.

13. A method as defined in claim 12 further comprising:

generating the mask of the bit-field in the machine word by:

calculating a difference between the extracted first and second portions of the bit-field definition number;

5 shifting a predetermined value by the calculated difference;

inverting the shifted predetermined value; and

shifting the inverted shifted predetermined value by the second portion of the bit-field definition number.

14. A method as defined in claim 12 further comprising:

reading a current value of the bit-field by:

ANDing the machine word with the generated mask to form an ANDed result having all the bits of the machine word outside of the bit-field

5 zeroed-out; and

shifting the ANDed result by the second portion of the bit-field definition number.

15. A method as defined in claim 12 further comprising:
- writing a write value to the bit-field by:
 - inverting the generated mask;
 - logically ANDing the machine word with the inverted generated mask
- 5 to form an ANDed result having the bits of the machine word only within the bit-field zeroed-out;
- shifting the write value by the second portion of the bit-field definition number; and
 - logically ORing the shifted write value with the ANDed result.